

J Optim Theory Appl (2014) 160:1020–1026  
DOI 10.1007/s10957-013-0336-2

---

# Evaluating Performance of Honey Bee Mating Optimization

Somayeh Karimi · Navid Mostoufi ·  
Rahmat Sotudeh-Gharebagh

Received: 22 February 2013 / Accepted: 6 May 2013 / Published online: 16 May 2013  
© The Author(s) 2013. This article is published with open access at Springerlink.com

**Abstract** The honey bee mating optimization (HBMO) algorithm is presented and tested with various test functions, and its performance is compared with the genetic algorithm (GA). It is shown that the HBMO algorithm can overcome the weaknesses of the GA. The HBMO converges faster than the GA. Even when the HBMO starts from a more improper initial condition than the GA, it can reach a better solution in a smaller number of function evaluations. Furthermore, in some cases, the GA was not able to reach the global minimum.

**Keywords** Honey bees mating optimization · Swarm-based algorithms · Swarm intelligence

## 1 Introduction

A branch of nature-inspired algorithms, known as swarm intelligence, is focused on insect behavior in order to develop some meta-heuristics which can initiate the insect's problem solution abilities. An ant colony [1, 2] and particle swarm optimization [3] are examples of the well-known algorithms that mimic insect behavior in problem modeling and solution. Honey bee mating algorithms (HBMO) belong to the novel swarm-based algorithms which are inspired by the marriage process in real bee colonies. The aim of this work is to study the performance of HBMO and its ability in finding a global optimum in comparison with other evolutionary algorithms like GA. In the present work, the HBMO algorithm is presented and applied to several test functions. Sensitivity analysis on crossover operators was performed and a

---

Communicated by David G. Hull.

S. Karimi · N. Mostoufi (✉) · R. Sotudeh-Gharebagh  
Process Design and Simulation Research Centre, School of Chemical Engineering,  
College of Engineering, University of Tehran, P.O. Box 11155/4563, Tehran, Iran  
e-mail: [mostoufi@ut.ac.ir](mailto:mostoufi@ut.ac.ir)

performance comparison of HBMO and GA was implemented using a benchmarking study.

## 2 Development of the HBMO Algorithm

There are five main steps in the development of the HBMO algorithm, as described below.

### Step 1: Mating flight of queen bees with drones

*In nature* In the marriage process, the queens mate during their mating flights far from the nest. A mating flight starts with a waggle dance performed by the queen that then starts a mating flight during which the drones follow the queen and mate with her in the air. In each mating, sperm reaches the spermatheca and accumulates there to form the genetic pool of the colony [4]. The queen is pursued by a large swarm of drones (drone comets) when copulation occurs. Insemination ends with the eventual death of the drone and the queen receiving the mating sign. The queen mates multiple times but the drone, inevitably, only once. These features make bee mating the most spectacular mating among insects [5].

*In algorithm* Each member of the bees colony, such as queen, drones and broods, are represented as a real coded string. A chromosome was used to represent a candidate solution to a problem where each gene in the chromosome represents a parameter of the candidate solution. The speed of each queen at the start of each mating flight is initialized at random. A list of drones is then randomly produced. The objective function for them is evaluated and the best one would be selected as the first queen. Next, a number of mating flights would be undertaken. The speed decreases during the mating flight. A drone mates with a queen probabilistically using an annealing function as follows [4]:

$$\text{prob}(Q, D) = \exp\left(\frac{-\Delta(f)}{S(t)}\right) \quad (1)$$

where  $\text{prob}(Q, D)$  is the probability of adding the sperm of drone  $D$  to the spermatheca of queen  $Q$  (that is, the probability of a successful mating),  $\Delta(f)$  is the absolute difference between the fitness of drone,  $f(D)$  and the fitness of queen,  $f(Q)$  and  $S(t)$  the speed of the queen at time  $t$ .

After each transition in space, the speed and energy of the queen decays according to the following equations:

$$S(t+1) = \alpha S(t) \quad (2)$$

$$E(t+1) = E(t) - \gamma \quad (3)$$

$$\gamma = 0.5 \frac{E(t_0)}{M} \quad (4)$$

where  $\alpha \in [0, 1]$ ,  $E(t)$  is the energy of the queen at time  $t$ ,  $\gamma$  is the amount of speed reduction after each transition and  $M$  is the maximum number of mating flight.

If the mating is successful (i.e., the drone passes the probabilistic decision rule), sperm of the drone is stored in the spermatheca of the queen. This part of the algorithm is like simulated annealing, i.e., a random number is produced and then a drone will pass the probabilistic decision rule if his probabilistic function value is greater than the random number. The spermatheca in developing the algorithm is simulated by a list of strings which belong to drones chromosome that have passed the probabilistic rule. The stopping criterion for mating flight of queen is reached when her spermatheca is full (maximum number of mating is reached), when her energy reaches a critical point or when her speed reaches its lower bound.

## Step 2: Creation of new broods by the queen

*In nature* Each time a queen lays fertilized eggs, she randomly retrieves a mixture of the sperm accumulated in the spermatheca to fertilize the egg [6].

*In algorithm* When all queens complete their mating flights, they start breeding and new broods are formed. Crossover operator is used to mix queen's genotype with drones. For a required number of broods, a queen is selected in proportion to her fitness and is mated with a randomly selected sperm from her spermatheca. This process is similar to that of the GA, with the difference that in the GA each offspring is born from two parents while in the HBMO algorithm a brood may have some genes from one drone and some from another. In other words, a brood does not have a specified father. Four crossover operators (intermediate, single point, two point, scattered) were considered in this work. Sensitivity analysis on the type of crossover was performed to find the most effective one.

## Step 3: Improvement of the broods' fitness by workers

*In nature* The function of workers is taking care of the brood and feeding them with the royal jelly. The royal jelly is a special food which belongs to the queen and makes the queen become bigger than other members in the hive. Feeding broods with royal jelly makes them become better and lets them have a potential of being the next queen.

*In algorithm* In the algorithm, this functionality of workers is modeled by representing them with a heuristic which acts to improve and/or take care of a set of broods. A set of different heuristics is represented by workers to improve the genotype of the broods. In this work, four types of mutation operator were used as the heuristic function.

## Step 4: Adaptation of the workers' fitness

*In nature* This step does not exist in nature.

*In algorithm* The rate of improvement in the brood's genotype, as a result of a heuristic application to that brood, is the definition of the fitness function for each worker. In this way, the fitness function of each worker becomes updated in every

iteration to give more chance to the worker which has more positive effect on the broods' gene. In the next iteration, the workers will be used in proportion to their fitness function. In the present study, the roulette wheel selection method was applied for this purpose.

### Step 5: Replacement of the least fit queen(s) with the fittest brood(s)

*In nature* Broods arise either from fertilized or unfertilized eggs. The former represent potential queens or workers, whereas the latter represent prospective drones [7].

*In algorithm* In the algorithm, broods cannot replace the worker but the best brood replaces the worst queen until there is no brood that is better than any of the queens. The remaining broods would not be killed but a predefined number of best broods (elites) would be selected and replaced with the worst ones. In this way, the list of drones will be updated in each mating flight and with this replacement the exploitation will be powered. A new mating flight begins until all assigned mating flights are completed or convergence criteria are met.

## 3 Genetic Algorithm

The principal ideas of the genetic algorithm (GA) were originally developed by Holland [8] and later refined and described in detail by Goldberg [9]. The GA is an adaptive method that imitates the genetic process in nature and can be applied to optimization problems. The main steps in the GA are as follows:

1. Encoding decision variables and placing them in chromosomes.
2. Creating an initial population.
3. Determining the fitness of every chromosome in the current population.
4. Selecting better chromosomes for mating.
5. Running crossover operators to generate new strings from the available chromosomes.
6. Performing mutation to randomly alter one or more genes.
7. Repeating steps 3–6 in order to find the best solutions.

## 4 Result and Discussion

Seven test functions, listed in Table 1, were used to evaluate the performance of the HBMO algorithm and compare it with that of the GA. Parameters of the HBMO and GA given in Tables 2 and 3, respectively, were selected after thorough testing. A number of different alternative values were tested in each test function and the ones selected are those that provided the best computational results. For some of the parameters, a range of values is given, which means that the used value is different for each test function.

Four crossover operators, mentioned above, were tested and the operator with the best result of HBMO and GA was used for each algorithm. Tables 4 and 5 represent

**Table 1** Benchmark functions

| Function No. | Function   | $D$ | Formulation   | Minimum   |
|--------------|------------|-----|---|-----------|
| 1            | Schaffer   | 2   | $f_1(x) = 0.5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$    | 0         |
| 2            | Sphere     | 100 | $f_2(x) = \sum_{i=1}^n x_i^2$   | 0         |
| 3            | Rosenbrock | 100 | $f_3(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$                          | 0         |
| 4            | Griewank   | 100 | $f_4(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$ | 0         |
| 5            | Rastrigin  | 100 | $f_5(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$                                   | 0         |
| 6            | Schwefel   | 100 | $f_6(x) = \sum_{i=1}^n [-x_i \sin(\sqrt{ x_i })]$   | -418.983n |
| 7            | Drop Wave  | 100 | $f_7(x) = -\frac{\cos(12\sqrt{\sum_{i=1}^n x_i^2}) + 1}{2 + 0.5(\sum_{i=1}^n x_i^2)}$       | -1        |

**Table 2** Values of the HBMO parameters used in this work

|   |                                      |
|---|--------------------------------------|
| Number of workers                         | 4                                    |
| Number of queens                          | 1                                    |
| Size of spermatheca                       | 35–45                                |
| Number of broods                          | 25–45                                |
| Maximum number of mating flight           | 20–500                               |
| Speed reduction ratio                     | 0.9                                  |
| Minimum speed                             | 0.001                                |
| Minimum energy                            | 0.0001                               |
| Mutation rate                             | 0.5–0.98                             |
| Weighting factor for crossover            | 0.98                                 |
| Lower and upper bound for mutation domain | Lower and upper bound of search area |
| Number of elites                          | 5–15                                 |

the objective functions in the sensitivity analysis on crossover operators in HBMO and GA, respectively. The best results corresponding to each test function for GA and HBMO are shown by bold numbers. It can be concluded from Table 4 that the intermediate crossover operator provides the best results for all test functions in the HBMO. However, Table 4 demonstrates that for the GA there are a variety of crossover types which can provide the best results. In the following, the best crossover was used for comparison.

Table 6 summarizes the comparison of results from GA and HBMO for the same number of function evaluations for the benchmark functions. Table 6 reveals that at the same number of function evaluations, the HBMO reaches a better minimum than the GA and is more likely not to be trapped in local minima. In the case of functions

**Table 3** Values of the GA parameters used in this work

|                    |   |
|--------------------|---|
| Population Size    | 20  |
| Selection Function | Stochastic uniform                            |
| Fitness scaling    | Rank  |
| Maximum Generation | 50–150  |
| Crossover Operator | Intermediate/Single point/Two point/Scattered |

**Table 4** Objective function values calculated in sensitivity analysis on crossover operators of HBMO

| Function No. | Intermediate           | Single point             | Two point                 | Scattered                 |
|--------------|------------------------|--------------------------|---------------------------|---------------------------|
| 1            | $8.93 \times 10^{-11}$ | $9.71705 \times 10^{-3}$ | $5.24 \times 10^{-2}$     | $3.754 \times 10^{-2}$    |
| 2            | $7.22 \times 10^{-28}$ | $2.49 \times 10^{-18}$   | $8.66 \times 10^{-8}$     | $8.66 \times 10^{-8}$     |
| 3            | $4.96 \times 10^{-9}$  | $2.472 \times 10^{-2}$   | $4.2249 \times 10^{-2}$   | $9.0227 \times 10^{-2}$   |
| 4            | $1.23 \times 10^{-13}$ | $8.86 \times 10^{-8}$    | $8.12188 \times 10^{-11}$ | $2.9246 \times 10^{-1}$   |
| 5            | $3.27 \times 10^{-13}$ | $1.73 \times 10^{-11}$   | $2.603 \times 10^{-8}$    | $4.16804 \times 10^{-11}$ |
| 6            | <b>−1247</b>           | −1070.034462             | 0                         | −1194.32432               |
| 7            | <b>−1</b>              | $9.35 \times 10^{-1}$    | $9.3618 \times 10^{-1}$   | $9.958 \times 10^{-1}$    |

**Table 5** GA objective function values using sensitivity analysis on crossover operators

| Function No. | Intermediate                             | Single point                             | Two point               | Scattered                                 |
|--------------|--|--|-------------------------|---|
| 1            | $9.7166 \times 10^{-3}$                  | <b><math>2.921 \times 10^{-4}</math></b> | $4 \times 10^{-4}$      | $9.716 \times 10^{-3}$                    |
| 2            | $5.24 \times 10^{-4}$                    | <b><math>9.137 \times 10^{-5}</math></b> | $1.577 \times 10^{-2}$  | $1.31 \times 10^{-3}$                     |
| 3            | <b><math>2.052 \times 10^{-3}</math></b> | $7.413 \times 10^{-2}$                   | $1.08 \times 10^{-2}$   | $1.191 \times 10^{-2}$                    |
| 4            | $6.86 \times 10^{-4}$                    | $4.572 \times 10^{-4}$                   | $6.672 \times 10^{-3}$  | <b><math>5.64 \times 10^{-5}</math></b>   |
| 5            | 2.023                                    | 2.613                                    | 1.989918                | <b><math>4.984 \times 10^{-2}</math></b>  |
| 6            | −11.84                                   | −11.84                                   | −11.84                  | <b>−11.84</b>                             |
| 7            | $-7.858 \times 10^{-1}$                  | $-9.362 \times 10^{-1}$                  | $-9.362 \times 10^{-1}$ | <b><math>-9.362 \times 10^{-1}</math></b> |

6 and 7 the GA cannot reach the global optimum point after 2000 and 700 function evaluations, respectively, and it becomes trapped in a local optimum.

## 5 Conclusions

Performance of the HBMO algorithm was investigated using seven benchmark functions. Results of the HBMO were compared with the results of the GA. We briefly recall the features of the HBMO:

- It was shown that the HBMO performs better than the GA in terms of speed of convergence and finding the global minimum.

**Table 6** Test function results

| Function No. | Bound           | Minimum value (Fmin) | Fmin (GA)                | Fmin (HBMO)            | Function evaluation |
|--------------|-----------------|----------------------|--------------------------|------------------------|---------------------|
| 1            | [−100, 100]     | 0                    | $2.921 \times 10^{-4}$   | $8.93 \times 10^{-11}$ | 1420                |
| 2            | [−5.12, 5.12]   | 0                    | $9.137 \times 10^{-5}$   | $7.22 \times 10^{-28}$ | 600                 |
| 3            | [−1.048, 3.048] | 0                    | $2.052 \times 10^{-3}$   | $4.96 \times 10^{-9}$  | 3000                |
| 4            | [−600, 600]     | 0                    | $5.64 \times 10^{-5}$    | $1.23 \times 10^{-13}$ | 750                 |
| 5            | [−5.12, 5.12]   | 0                    | $4.984 \times 10^{-2}$   | $3.27 \times 10^{-13}$ | 850                 |
| 6            | [−500, 500]     | −1256.9              | −11.8359                 | −1247                  | 2000                |
| 7            | [−5.12, 5.12]   | −1                   | $-9.3625 \times 10^{-1}$ | −1                     | 700                 |

- The optimization results demonstrated that the HBMO algorithm is convergent to the global minimum for all the test functions, although these test functions are complex and have many local optima.
- The results are promising and encourage further research for applying the HBMO algorithm to complex and real-time optimization problems.

**Acknowledgements** The authors gratefully acknowledge the support (Grant No. 87040013) from Iran National Science Foundation (INSF).

**Open Access** This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

## References

1. Dorigo, M.: Ant colony system: a cooperative learning approach to the travelling salesman problem. *IEEE Trans. Evol. Comput.* **1**, 53–66 (1997)
2. Dorigo, M.: Ant system: optimization by a colony of cooperation agents. *IEEE Trans. Syst. Man Cybern.* **26**, 29–41 (1996)
3. Kennedy, J.: The particle swarm: social adaptation of knowledge. In: *Proceedings of IEEE International Conference on Evolutionary Computation, IEC'97, USA* (1997)
4. Abbass, H.: A monogenous MBO approach to satisfiability. In: *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA'2001, Las Vegas, NV, USA* (2000)
5. Abbass, H.: Marriage in honey-bee optimization (MBO): a haplometrosis polygynous swarming approach. In: *Proceedings of the Congress on Evolutionary Computation (CEC'2001), Seoul, Korea* (2001)
6. Page, R.: The evolution of multiple mating behavior by honey bee queens (*Apis mellifera* L.). *Genetics* **96**, 263–273 (1980)
7. Afshar, A.: Honey-bee mating optimization (HBMO) algorithm for optimal reservoir operation. *J. Franklin Inst.* **344**, 452–462 (2007)
8. Holland, J.: *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, Ann Arbor (1975)
9. Goldberg, D.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, New York (1989)